

BTS SIO 2022

Support et mise à disposition de services informatiques (E4)

PAGE DE PRÉSENTATION DU DOSSIER

N° de candidat : | 0 | 2 | 1 | 4 | 6 | 7 | 1 | 7 | 4 | 7 | 3 |

NOM : DUFOUR

Prénom : Thomas

Date de passage ¹ : 24 / 05 /2022	Heure de passage ¹ : 9h30
--	--------------------------------------

CATEGORIE CANDIDAT ² (UNE CASE A COCHER)	
☐ Scolaire ☒ Apprenti ☐ Formation professionnelle continue ☐ Expérience professionnelle 3 ans	☐ Ex-scolaire ☐ Ex-apprenti ☐ Ex-formation professionnelle continue

¹ Informations communiquées sur votre convocation envoyée en mars-avril 2022

² Informations communiquées sur votre confirmation d'inscription

Tampon de
l'établissement



BTS SIO – Dossier Etudiant
Justificatif d'acquisition des compétences

Rédacteur(s)	Version	Date	Nb pages
Heilig Fabian	1.1	15/03/2022	10

Epreuve E4
Support et mise à disposition de
services informatiques

SOMMAIRE

- 1 INTRODUCTION 4
- 2 MISSION 3 MISE EN PLACE D’UN SERVEUR WEB 5
 - 2.1 Cahier des charges..... 5
 - 2.2 Etude et conception de la solution..... 6
 - 2.3 Gestion de projet 7
 - 2.4 Mise en œuvre..... 8
 - 2.5 Bilan..... 12

1 Introduction

L'objectif de ce document est de vous présenter les missions professionnelles que j'ai effectué dans le cadre de ma formation BTS SIO à l'école IRIS de Strasbourg.

Ces missions peuvent être de trois types :

- Effectuées en entreprise durant une alternance
- Effectuées en stage en entreprise
- Effectuées à l'école (compte-rendu de TP, projet collaboratif)

Le type de la mission sera précisé dans chaque cahier des charges.

Ce document se compose des parties suivantes :

Chapitres	Contenu
Chapitres 2 à 8	Présentation des missions, avec pour chacune : - Le cahier des charges - La solution proposée - La gestion de projet - La mise en œuvre - Le bilan du projet

2 Mission 3 : Mise en place d'un serveur Web

2.1 Cahier des charges

Type de mission
Mission effectuée en alternance.
Contexte
La société Wendling Adhésif devant la charge de calcul toujours plus haute et avec un serveur très peu performant, décide de changer de serveur afin de parfaire son parc informatique et sa fluidité de travail. L'ancien serveur hébergeait l'application web (ERP) interne de l'entreprise, il faut donc que le nouveau serveur puisse l'héberger. L'OS Ubuntu est déjà présent sur ce serveur.
Demande du client
Mettre en place un serveur web pouvant héberger l'application web de l'entreprise
Expression du besoin
- Installation d'apache - Installation de PHP - Installation d'un serveur MySQL
Budget disponible
Aucun budget nécessaire pour cette tâche
Outils disponibles
Une machine avec Ubuntu d'installé
Une connexion en SSH
Contraintes
La contrainte majeure de cette mission a été l'installation en général des services web. En effet, n'ayant jamais installé de serveur Web en production jusqu'à maintenant et n'ayant pas utilisé l'interface Linux depuis un moment, il a fallu que je me forme sur le sujet.

2.2 Etude et conception de la solution

2.2.1 La solution imaginée

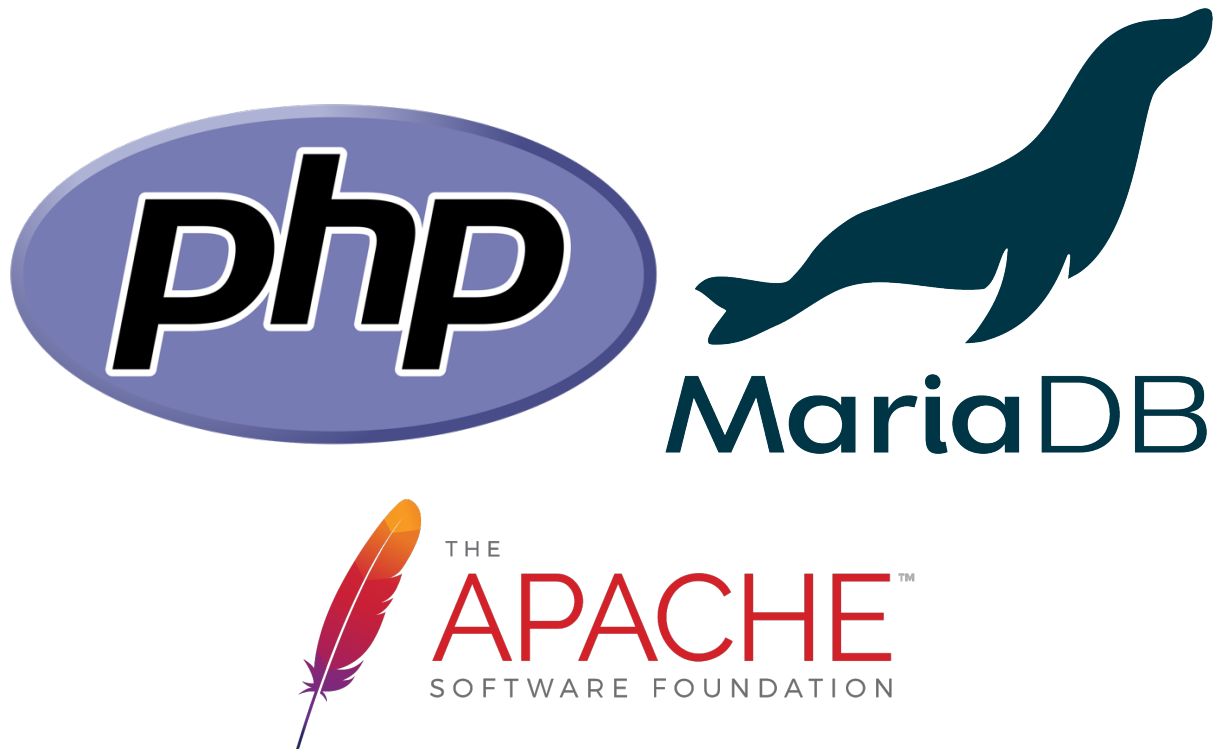
Pour répondre au besoin de cette mission il a été décidé d'installer un serveur LAMP (Linux Apache MySQL Php).

C'est une structure web qui est assez connue puisque déjà utilisée sur l'ancien serveur. Le site contenant un fichier « .htaccess » et la structure de fichier déjà connue, il a été décidé de garder le serveur HTTP Apache.

Il est aussi prévu de mettre des restrictions IP au sein du Virtual Host afin d'augmenter la sécurité concernant la connexion au serveur Web.

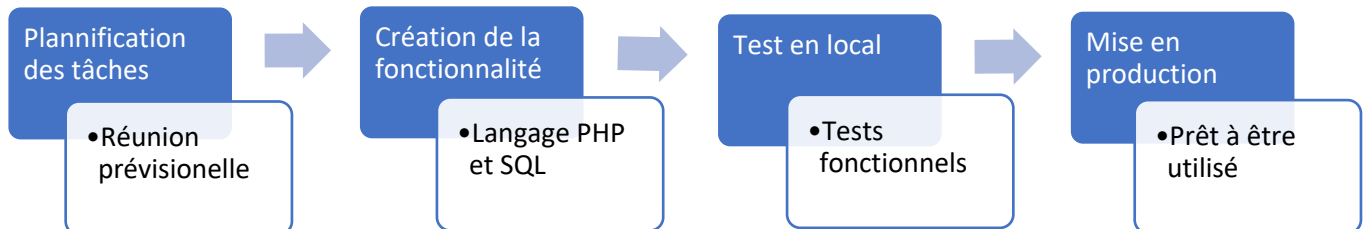
Étant donné que le langage utilisé pour le développement de l'application web est PHP, il nous a fallu installer PHP sur le serveur. La version de PHP utilisée au sein de l'ancien serveur est PHP 7.4, il nous faut donc choisir la version de PHP 7.4.

Concernant l'installation de la base de données, il a fallu choisir entre MySQL et Maria DB. Le choix s'est plutôt concentré vers Maria DB étant donné l'approche Open Source du système de gestion de base de données. Pour cela, il faudra installer le serveur et le client de Maria DB.



2.3 Gestion de projet

2.3.1 Planing de déploiement de la solution



2.3.2 Budget

Pas de budget alloué pour cette opération.

2.4 Mise en œuvre

2.1 Procédure d'installation

I- Prérequis

Pour commencer l'installation des éléments composants un serveur Web, il nous faut checker les mises à jours disponibles avec la commande :

```
# apt update
```

Une fois les mises à jours recherchés, il nous faut installer les mises à jour avec la commande :

```
# apt upgrade
```

Une fois ces commandes effectuées, nous pouvons commencer nos installations dans les meilleures conditions possibles.

II- Installation d'Apache

Nous pouvons donc commencer l'installation d'Apache sur la machine. Sudo étant déjà installé, il nous faut taper la commande suivante afin d'initialiser l'installation d'Apache :

```
# sudo apt install -y apache2 apache2-utils
```

Cette commande nous permet donc d'installer apache2 ainsi que le paquet apache2-utils qui fournit des programmes complémentaires tels que :

- Ab (outil de mesure de performances Apache)
- Htdbm (manipuler les fichiers d'authentification basique au format DBM)
- ...

Juste après avoir installé Apache 2, le service devrait redémarrer automatiquement. Si ce n'est pas le cas, tapez la commande suivante :

```
# sudo systemctl restart apache2.services
```

Juste après avoir redémarrer les services, nous pouvons taper cette commande afin de vérifier l'installation d'apache :

```
# systemctl status apache2
```

Nous devrions avoir un affichage comme ceci :


```
thomas@Server-Thomas:~$ systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
   Active: active (running) since Tue 2022-04-05 09:20:34 UTC; 3 weeks 3 days ago
     Docs: https://httpd.apache.org/docs/2.4/
  Process: 33496 ExecReload=/usr/sbin/apachectl graceful (code=exited, status=0/SUCCESS)
 Main PID: 21457 (apache2)
    Tasks: 55 (limit: 38359)
   Memory: 7.4M
   CGroup: /system.slice/apache2.service
           └─21457 /usr/sbin/apache2 -k start
           └─33502 /usr/sbin/apache2 -k start
           └─33503 /usr/sbin/apache2 -k start
```

Nous pouvons ensuite lancer le serveur apache avec cette commande :

```
# systemctl start apache2
```

Puis nous pouvons définir qu'apache se lance dès le lancement de la machine avec :

```
# systemctl enable apache2
```

Nous vérifions donc la version d'Apache avec :

```
# apache2 -v
```

```
Server version: Apache/2.4.46 (Ubuntu)
Server built:   2022-01-05T14:38:48
```

Après avoir vérifié ça, il nous faut vérifier qu'apache est bien en route en tapant l'IP du serveur dans un navigateur web qui devrait nous retourner cette page :



Apache2 Ubuntu Default Page

ubuntu

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in `/usr/share/doc/apache2/README.Debian.gz`**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```

/etc/apache2/
|-- apache2.conf
    |-- ports.conf
|-- mods-enabled
    |-- *.load
    |-- *.conf
|-- conf-enabled
    |-- *.conf
|-- sites-enabled
    |-- *.conf

```

- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.

III- Installation de mariaDb

Passons maintenant à l'installation de mariaDb sur le serveur, pour lancer l'installation on tape la commande suivante :

```
# sudo apt install mariadb-server mariadb-client
```

Une fois le client et le serveur installé, on vérifie le statut

```
# statut systemctl mariadb
```

```
● mariadb.service - MariaDB 10.5.13 database server
   Loaded: loaded (/lib/systemd/system/mariadb.service; enabled; vendor preset: enabled)
   Active: active (running) since Fri 2022-04-29 10:54:41 UTC; 12s ago
     Docs: man:mariadb(8)
           https://mariadb.com/kb/en/library/systemd/
 Process: 34369 ExecStartPre=/usr/bin/install -m 755 -o mysql -g root -d /var/run/mysql (code=exited, status=0/SUCCESS)
 Process: 34372 ExecStartPre=/bin/sh -c systemctl unset-environment _WSREP_START_POSITION (code=exited, status=0/SUCCESS)
 Process: 34375 ExecStartPre=/bin/sh -c [ ! -e /usr/bin/galera_recovery ] && VAR= || VAR='cd /usr/bin/..; /usr/bin/galera_recovery'; [ $? -eq 0 ] && systemctl set-e
 Process: 34457 ExecStartPost=/bin/sh -c systemctl unset-environment _WSREP_START_POSITION (code=exited, status=0/SUCCESS)
 Process: 34459 ExecStartPost=/etc/mysql/debian-start (code=exited, status=0/SUCCESS)
 Main PID: 34445 (mariabdd)
   Status: "Taking your SQL requests now..."
    Tasks: 12 (limit: 38359)
  Memory: 70.7M
   CGroup: /system.slice/mariadb.service
           └─34445 /usr/sbin/mariabdd
```

On lance la base de donnée MariaDb :

```
# sudo systemctl start mariadb
```

Puis on l'active au lancement de la machine :

```
# sudo systemctl enable mariadb
```

Les manipulations suivantes servent à configurer MariaDB, pour commencer on tape la commande afin de définir un mot de passe pour l'utilisateur root :

```
thomas@Server-Thomas:~$ sudo mysql_secure_installation
[sudo] password for thomas:

NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MariaDB
SERVERS IN PRODUCTION USE! PLEASE READ EACH STEP CAREFULLY!

In order to log into MariaDB to secure it, we'll need the current
password for the root user. If you've just installed MariaDB, and
haven't set the root password yet, you should just press enter here.

Enter current password for root (enter for none):
```

On peut voir ici que la base de données est bel et bien installée :

```
thomas@Server-Thomas:~$ mariadb --version
mariadb Ver 15.1 Distrib 10.5.13-MariaDB, for debian-linux-gnu (x86_64) using EditLine wrapper
```

IV- Installation de PHP

Pour commencer il nous faut installer PHP 7.4 étant donné que le site web de l'entreprise est développé avec PHP 7.4

```
# sudo apt install php7.4 libapache2-mod-php7.4 php7.4-mysql php-common  
php7.4-cli php7.4-common php7.4-json php7.4-opcache php7.4-readline
```

On active ensuite le module Apache avant de le redémarrer.

```
# sudo a2enmod php7.4  
# sudo systemctl restart apache2
```

On peut ensuite vérifier la version de php avec :

```
# php --version
```

```
thomas@Server-Thomas:~$ php --version  
PHP 7.4.16 (cli) (built: Nov 25 2021 21:36:47) ( NTS )  
Copyright (c) The PHP Group  
Zend Engine v3.4.0, Copyright (c) Zend Technologies  
with Zend OPcache v7.4.16, Copyright (c), by Zend Technologies
```

Nous pouvons dès à présent mettre nos fichiers php dans le répertoire suivant :

```
# /var/www/html/
```

V- Restrictions IP

Pour faire une restriction IP, il suffit de se rendre dans le Virtual Host situé dans :

```
# /etc/apache2/sites-available/nomdedomaine.conf
```

```
<VirtualHost *:80>  
    # The ServerName directive sets the request scheme, hostname and port that  
    # the server uses to identify itself. This is used when creating  
    # redirection URLs. In the context of virtual hosts, the ServerName  
    # specifies what hostname must appear in the request's Host: header to  
    # match this virtual host. For the default virtual host (this file) this  
    # value is not decisive as it is used as a last resort host regardless.  
    # However, you must set it for any further virtual host explicitly.  
    #ServerName www.example.com  
    Require ip ***.***.**.*/**  
    ServerAdmin webmaster@localhost  
    DocumentRoot /var/www/html
```

2.2 Bilan

2.2.1 Validation des exigences point par point

Le serveur LAMP a bien été mis en place avec les restrictions ip des utilisateurs de l'application web.

- Installation d'apache : **FAIT**
- Installation de PHP : **FAIT**
- Installation d'un serveur MySQL : **FAIT**

2.2.2 Axes d'amélioration

Une amélioration qui pourrait être envisageable serait d'installer un certificat SSL sur le site afin de pouvoir faire appel à des API comme celles d'Ebay. Un service qui pourrait aussi être installé est Fail2Ban permettant de ban les personnes mettant trop d'essai pour réussir à se connecter au serveur.

2.2.3 Compétences acquises

J'ai acquis lors de cette mission de nombreuses compétences concernant le système. Je peux dès à présent monter un serveur Web de A à Z et réussir à comprendre les erreurs potentiels que je pourrais avoir lors de ces installations.